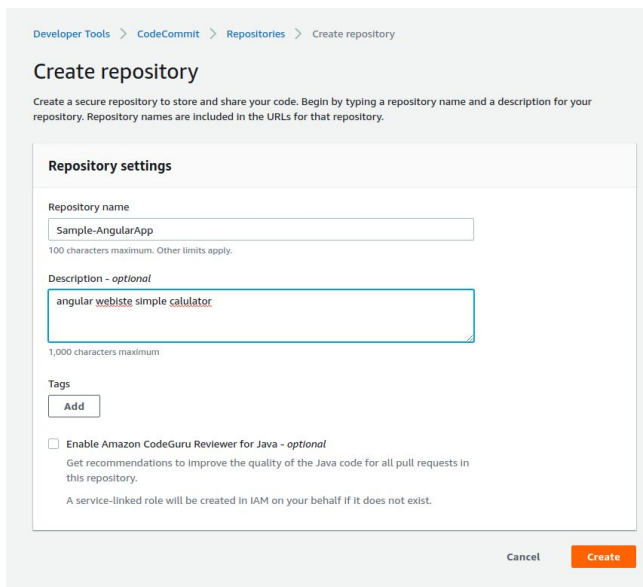


## Dokumentasi Blue-Green Deployment dengan CodeDeploy CI/CD Code Commit

Untuk melakukan proses deployment blue green dengan codedeploy dan ci/cd code commit ini ada beberap ketentuan :

1. Akun AWS - Free Tier
2. Login menggunakan IAM User
3. Saya menggunakan Website berbasis Angular sebagai contoh

1. Membuat repository pada codecommit di mana ini akan menjadi version control sekaligus menjadi tempat atau source code yang telah di siapakan :



Developer Tools > CodeCommit > Repositories > Create repository

### Create repository

Create a secure repository to store and share your code. Begin by typing a repository name and a description for your repository. Repository names are included in the URLs for that repository.

#### Repository settings

Repository name

100 characters maximum. Other limits apply.

Description - optional

1,000 characters maximum

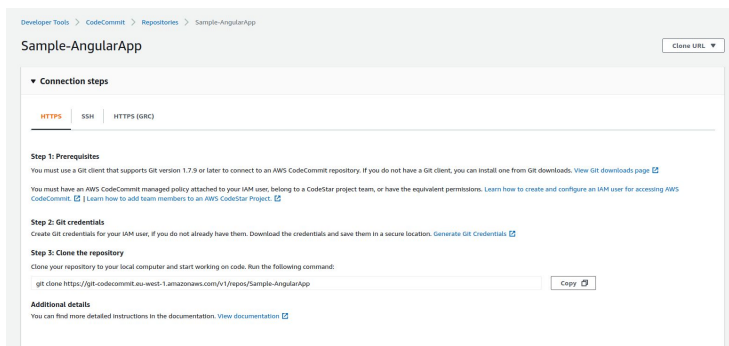
Tags

☐ Enable Amazon CodeGuru Reviewer for Java - optional

Get recommendations to improve the quality of the Java code for all pull requests in this repository.

A service-linked role will be created in IAM on your behalf if it does not exist.

Naman dan deskripsi Repository pada di codecommit  
Lalu klik **create**



Developer Tools > CodeCommit > Repositories > Sample-AngularApp

### Sample-AngularApp

▼ Connection steps

HTTPS | SSH | HTTPS (SBC)

**Step 1: Prerequisites**

You must use a Git client that supports Git version 1.7.9 or later to connect to an AWS CodeCommit repository. If you do not have a Git client, you can install one from [Git downloads](#). [View Git downloads page](#)

You must have an AWS CodeCommit managed policy attached to your IAM user, belong to a CodeStar project team, or have the equivalent permissions. [Learn how to create and configure an IAM user for accessing AWS CodeCommit](#). [Learn how to add team members to an AWS CodeStar Project](#)

**Step 2: Git credentials**

Create Git credentials for your IAM user, if you do not already have them. Download the credentials and save them in a secure location. [Generate Git Credentials](#)

**Step 3: Clone the repository**

Clone your repository to your local computer and start working on code. Run the following command:

```
git clone https://git-codecommit.eu-west-1.amazonaws.com/v1/repos/Sample-AngularApp
```

**Additional details**

You can find more detailed instructions in the documentation. [View documentation](#)

Selanjutnya terlihat url untuk clone repository dan push dari local computer

Untuk mendapatkan Username dan Password Code Commit masuk ke  
IAM>User>Securitycredential

HTTPS Git credentials for AWS CodeCommit

Generate a user name and password you can use to authenticate HTTPS connections to AWS CodeCommit repositories. You can generate and store up to 2 sets of credentials. [Learn more](#)

	User name	Status	Created
<input type="radio"/>	vvn-at-837881029860	Active	2020-07-11 22:20 UTC+0700

Credentials for Amazon Keyspaces (for Apache Cassandra)

Lalu generate HTTPS Git Credential for AWS CodeCommit

Clone repository yg tela di buat sebelumnya

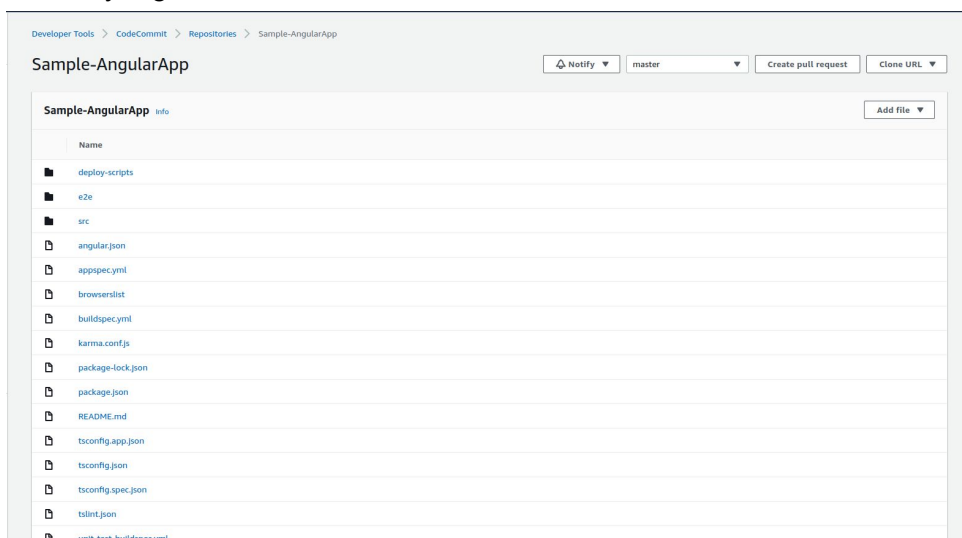
```
vvn@Skinnyboys:~/Documents/aws$ git clone https://git-codecommit.eu-west-1.amazonaws.com/v1/repos/Sample-AngularApp
Cloning into 'Sample-AngularApp'...
Username for 'https://git-codecommit.eu-west-1.amazonaws.com': vvn-at-837881029860
Password for 'https://vvn-at-837881029860@git-codecommit.eu-west-1.amazonaws.com':
warning: You appear to have cloned an empty repository.
vvn@Skinnyboys:~/Documents/aws$
```

Lalu copy file website angular yg akan di deploy nanti

Setelah selesai di copy selanjutnya push file website ke repository code commit

```
vvn@Skinnyboys:~/Documents/aws/Sample-AngularApp$ git push origin master
Counting objects: 47, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (43/43), done.
Writing objects: 100% (47/47), 130.59 KiB | 3.96 MiB/s, done.
Total 47 (delta 0), reused 0 (delta 0)
To https://git-codecommit.eu-west-1.amazonaws.com/v1/repos/Sample-AngularApp
 * [new branch]      master -> master
vvn@Skinnyboys:~/Documents/aws/Sample-AngularApp$
```

Jika sudah selesai pastikan file telah terupload dengan benar buka repository pada code commit yang ada di AWS




Terlihat file nya sudah terupload


## Create IAM Role untuk CodeDeploy Agent


Create role


1234

Select type of trusted entity

**AWS service**  
EC2, Lambda and others

**Another AWS account**  
Belonging to you or 3rd party

**Web identity**  
Cognito or any OpenID provider

**SAML 2.0 federation**  
Your corporate directory

Allows AWS services to perform actions on your behalf. [Learn more](#)

Choose a use case

Common use cases

**EC2**  
Allows EC2 instances to call AWS services on your behalf.

**Lambda**  
Allows Lambda functions to call AWS services on your behalf.

Or select a service to view its use cases

API Gateway	CodeGuru	ElastiCache	Kinesis	RoboMaker
AWS Backup	CodeStar Notifications	Elastic Beanstalk	Lake Formation	S3
AWS Chatbot	Comprehend	Elastic Container Service	Lambda	SMS
AWS Support	Config	Elastic Transcoder	Lex	SNS
Amplify	Connect	ElasticLoadBalancing	License Manager	SWF
AppStream 2.0	DMS	Forecast	Machine Learning	SageMaker
AppSync	Data Lifecycle Manager	GameLift	Macie	Security Hub
Application Auto Scaling	Data Pipeline	Global Accelerator	Managed Blockchain	Service Catalog
Application Discovery Service	DataSync	Glue	MediaConvert	Step Functions
Batch	DeepLens	Greengrass	Migration Hub	Storage Gateway
	Directory Service	GuardDuty	OpsWorks	Systems Manager

\* Required

CancelNext: Permissions

## Klik AWS Service > EC2 > Next

Roles > CodeDeployEC2ServiceRole2

Summary

Delete role

Policy AWSCodeDeployFullAccess has been attached for the CodeDeployEC2ServiceRole2.

Role ARN

arn:aws:iam:837881029860:role/CodeDeployEC2ServiceRole2

Role description

Allows CodeDeploy to call AWS services such as Auto Scaling on your behalf. | Edit

Instance Profile ARNs

Path

/

Creation time

2020-07-13 01:45 UTC+0700

Last activity

2020-07-13 14:18 UTC+0700 (Today)

Maximum CLI/API session duration

1 hour Edit

Permissions

Trust relationships

Tags

Access Advisor

Revoke sessions

Permissions policies (3 policies applied)

Attach policies

Add inline policy

Policy name	Policy type	
AutoScalingFullAccess	AWS managed policy	x
AWSCodeDeployFullAccess	AWS managed policy	x
AdministratorAccess	AWS managed policy	x
Permissions boundary (not set)		

Ini hasil pembuatan IAM Rule untuk Webserver

Selanjutnya create Instance Amazonlinux2 dan install CodeDeploy Agent, dan nginx untuk webserver

**Step 3: Configure Instance Details**  
Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

**Number of instances** ⓘ 1 [Launch into Auto Scaling Group](#) ⓘ

**Purchasing option** ⓘ ☐ Request Spot instances

**Network** ⓘ vpc-7c0ff605 (default) [Create new VPC](#)

**Subnet** ⓘ No preference (default subnet in any Availability Zone) [Create new subnet](#)

**Auto-assign Public IP** ⓘ Use subnet setting (Enable) ⓘ

**Placement group** ⓘ ☐ Add instance to placement group

**Capacity Reservation** ⓘ Open [Create new Capacity Reservation](#)

**IAM role** ⓘ webservers2 [Create new IAM role](#)

**Shutdown behavior** ⓘ Stop ⓘ

**Stop - Hibernate behavior** ⓘ ☐ Enable hibernation as an additional stop behavior

**Enable termination protection** ⓘ ☐ Protect against accidental termination

**Monitoring** ⓘ ☐ Enable CloudWatch detailed monitoring  
[Additional charges apply.](#)

**Tenancy** ⓘ Shared - Run a shared hardware instance ⓘ  
[Additional charges may apply when launching Dedicated instances.](#)

**Elastic Inference** ⓘ ☐ Add an Elastic Inference accelerator  
[Additional charges apply.](#)

**T2/T3 Unlimited** ⓘ ☐ Enable  
[Additional charges may apply](#)

Pilih Role yang telah di buat tadi

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

### Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more about Amazon EC2 security groups.](#)

Assign a security group: ☒ Create a new security group ☐ Select an existing security group

Security group name:

Description:

Type ①	Protocol ①	Port Range ①	Source ①	Description ①
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Custom 0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop

[Add Rule](#)

**Warning**

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

[Cancel](#) [Previous](#) [Review and Launch](#)

[Feedback](#) [English \(US\)](#) © 2009 - 2020 Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

Create Scurity Group dan tambahkan Port 80 untuk HTTP  
Lalu Launch Instance

Connet melalui SSH untuk melakukan Installasi CodeDeploy Agent dan Nginx

```
Installed:
  codedeploy-agent.noarch 0:1.1.0-4

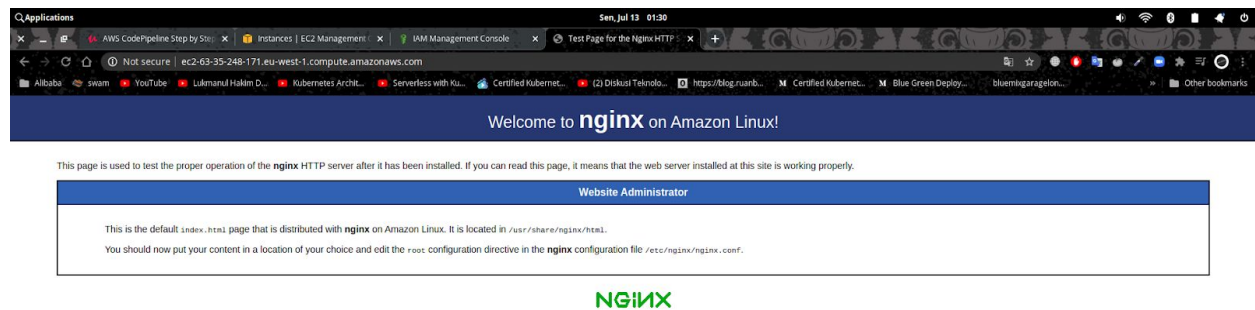
Complete!
I, [2020-07-12T18:26:12.428580 #22084] INFO -- : Update check complete.
I, [2020-07-12T18:26:12.428666 #22084] INFO -- : Stopping updater.
[ec2-user@ip-172-31-47-137 ~]$ sudo service codedeploy-agent status
The AWS CodeDeploy agent is running as PID 22163
[ec2-user@ip-172-31-47-137 ~]$
```

Update, install Ruby, Download CodeDeploy Agent dan Install Agent  
Dan cek apakah sudah terinstall

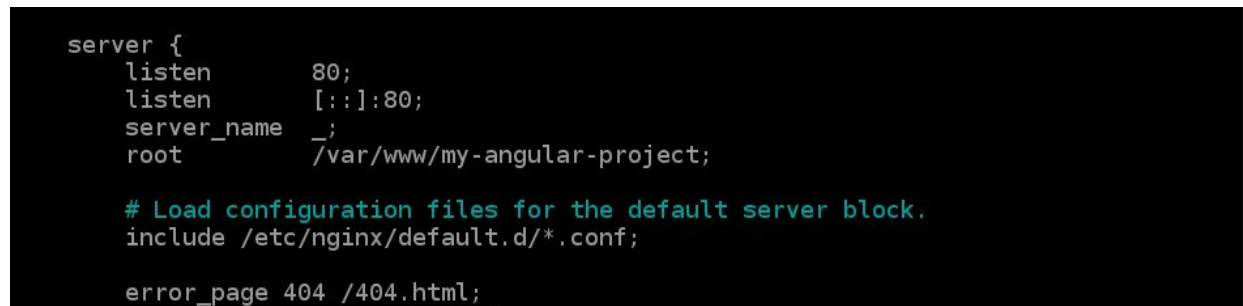
Install Nginx dan aktifkan

```
Redirecting to /bin/systemctl status nginx.service
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled; vendor preset: disabled)
   Active: active (running) since Sun 2020-07-12 18:29:05 UTC; 5s ago
     Process: 22398 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
     Process: 22395 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
     Process: 22394 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
   Main PID: 22401 (nginx)
     CGroup: /system.slice/nginx.service
             └─22401 nginx: master process /usr/sbin/nginx
               └─22402 nginx: worker process
```

Patikan dengan cek DNS pada Browser



Buat Folder untuk Deployment dan pada file config nginx arahkan folder root ke folder deployment yang telah di buat sebelumnya



Restart Nginx dan pastikan di browser tidak tampil apa-apa karena di folder deployment kosong



Jika sudah berjalan Instance untuk CodeDeploy Agent- maka IAMs sudah terbuat instance yang di buat tadi bisa di matik/stop untuk menghemat cost,

Create AMIs Image yang nantinya di gunakan untuk pembuatan Template Autoscaling pada Launch Cofiguration

Pilih Intance yang telah di buat sebelumnya > Action > Image > Create Image



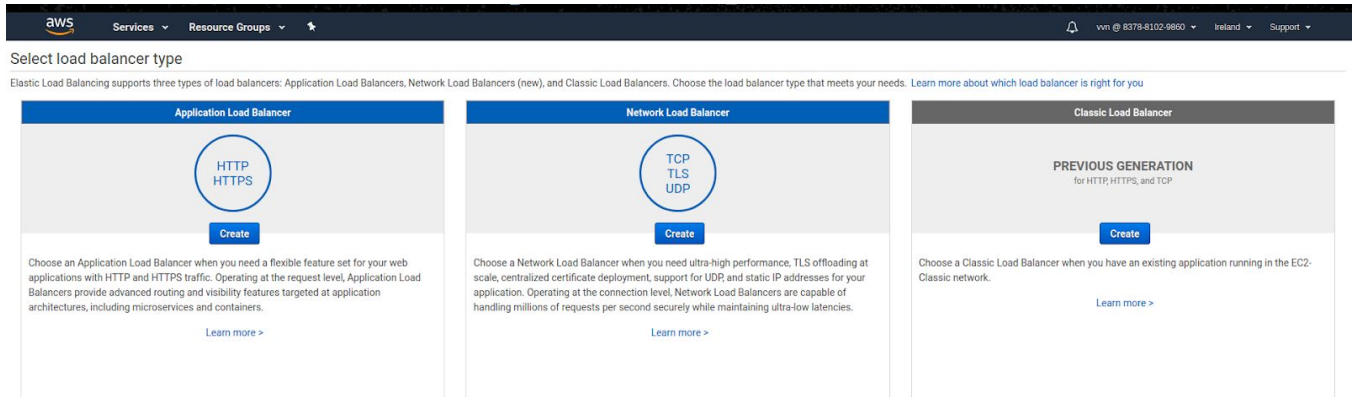
Buat nama Image untuk AMs > isi deskripsi > create mage

Sekarang Create Role untuk CodeDeploy

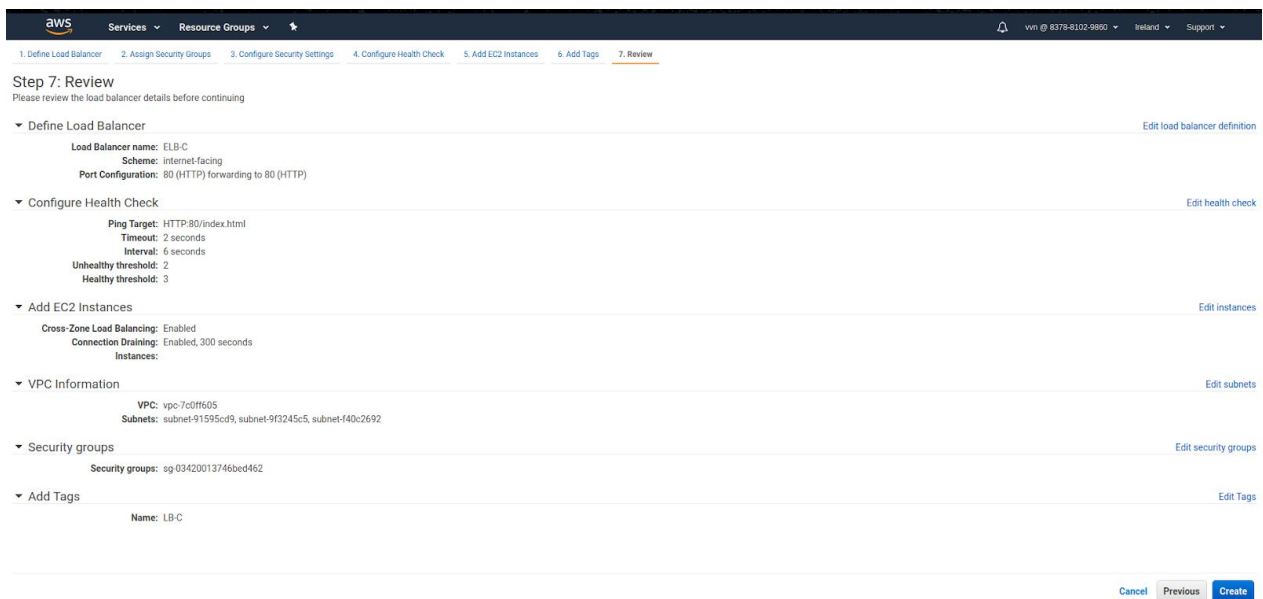
Pilih AWS Service > CodeDeploy > CodeDeploy > Next

Beri nama role lalu create

Selanjutnya Cretae LoadBalancer



## Pilih Classic LoadBalancer



## Klik Create

## Setelah selesai membuat LoadBalancer

## Selanjutnya buat konfigurasi untuk Autoscaling

### Create LaunchConfiguration

Tetapi di sini saya menggunakan template instance - konfigurasi ini nanti akan menentukan jenis instance yang akan di pakai untuk proses deployment sesuai dengan basic instance IAMs yang di buat sebelumnya

Klik LaunchTemplate>Create LaunchTemplate > isi Nama dan Deskripsi >



**Amazon machine image (AMI)** [Info](#)

AMI

web-server-AMI2  
ami-0f57f5390bfd87641  
Catalog: My AMIs architecture: 64-bit (x86) virtualization: hvm

**Instance type** [Info](#)

Instance type

t2.micro  
Family: General purpose 1 vCPU 1 GiB Memory Free tier eligible [Instance types](#)

**Key pair (login)** [Info](#)

Key pair name

aws-ec2 [Create new key pair](#)

Untuk AMI Pilih Image yang telah di buat sebelumnya > instance type pilih t2micro > keypair gunakan yang telah ada

**▼ Advanced details** [Info](#)

Purchasing option [Info](#)

☐ Request Spot Instances  
Request Spot Instances at the Spot price, capped at the On-Demand price

IAM instance profile [Info](#)

webserv2  
arn:aws:iam::837881029860:instance-profile/webserv2 [Create new IAM profile](#)

Shutdown behavior [Info](#)

Don't include in launch template

Stop - Hibernate behavior [Info](#)

Don't include in launch template

Termination protection [Info](#)

Don't include in launch template

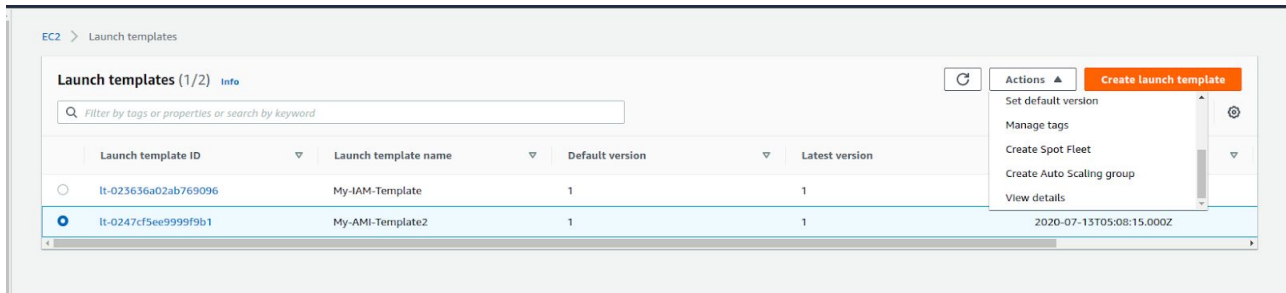
Detailed CloudWatch monitoring [Info](#)

Disable

Untuk IAM Profile gunakan atau sama dengan nama security group yang di buat di awal  
Lalu klik create

Selanjutnya Create Autoscaling Group

Pilih template yang telah di buat tadi > Action > Create Auto Scaling Group



Isikan nama untuk Group, pada tempelate terlihat template yang di buat sebelumnya lalu next

**Name**

Auto Scaling group name  
Enter a name to identify the group.

My-Auto-Scaling-Group-2

Must be unique to this account in the current Region and no more than 255 characters.

---

**Launch template** [info](#) [Switch to launch configuration](#)

**Launch template**  
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

My-AMI-Template2

[Create a launch template](#)

**Version**

Default (1)

[Create a launch template version](#)

<p>Description</p> <p>Template Instance untuk di gunakan Autoscaling Group</p> <p>AMI ID</p> <p>ami-0f57f5390bfd87641</p> <p>Key pair name</p> <p>aws-ec2</p>	<p>Launch template</p> <p><a href="#">My-AMI-Template2</a> lt-0247cf5ee9999f9b1</p> <p>Security groups</p> <p>-</p>	<p>Instance type</p> <p>t2.micro</p> <p>Security group IDs</p> <p><a href="#">sg-03420013746bed462</a></p>
---	---	--

Pada Cofigurasi

Untuk Purchase biarka setingan Default

Pada Networking pilih VPC yang tersedia > subnet pilih semua subnet yang tersedia > next

### Purchase options and instance types [Info](#)

☒ **Adhere to launch template**  
The launch template determines the purchase option (On-Demand or Spot) and instance type.

☐ **Combine purchase options and instance types**  
Specify how much On-Demand and Spot capacity to launch and multiple instance types (optional). This choice is most helpful for optimizing the scale and cost for a fleet of instances.

### Network [Info](#)

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

**VPC**

vpc-7c0ff605  
172.31.0.0/16    Default

↻

[Create a VPC](#)

**Subnets**

Select subnets

↻

eu-west-1a | subnet-91595cd9  
172.31.16.0/20    Default

eu-west-1b | subnet-9f3245c5  
172.31.32.0/20    Default

eu-west-1c | subnet-f40c2692  
172.31.0.0/20    Default

[Create a subnet](#)

## Pada Advance Configuraton

Untuk Loadbalancing klik Classic Load balancer > pilih loadbalancer yang di buat sebelumnya

Pada heatlh chek centang ELB dan seting health chek priode menjadi 200

Ini akan menseting atau mengecek apakah instanc bermalasaah atau tidak selama 200 detik pada proses deployment

Lalu next

**Load balancing - optional** [info](#)

☒ Enable load balancing

☐ Application Load Balancer or Network Load Balancer
 ☒ Classic Load Balancer

Choose a load balancer

Select load balancer ▼ ↺

ELB-C ✕

[Create a load balancer](#)

---

**Health checks - optional**

**Health check type** [info](#)

EC2 Auto Scaling automatically replaces instances that fail health checks. If you enabled load balancing, you can enable ELB health checks in addition to the EC2 health checks that are always enabled.

☐ EC2
 ☒ ELB

**Health check grace period**

The amount of time until EC2 Auto Scaling performs the first health check on new instances after they are put into service.

200 ↕ seconds

---

**Additional settings - optional**

**Monitoring** [info](#)

☐ Enable group metrics collection within CloudWatch

Cancel
 Previous
 Skip to review
 Next

## Pada Konfigurasi Group

Ini kan menyetting berapa jumlah instance yang akan di buat dan maksimal serta minimal instance

Desired capacity > 3

Minimum capacity > 1

Maximum capacity > 4

**Configure group size and scaling policies** [info](#)

Set the desired, minimum, and maximum capacity of your Auto Scaling group. You can optionally add a scaling policy to dynamically scale the number of instances in the group.

**Group size - optional** [info](#)

Specify the size of the Auto Scaling group by changing the desired capacity. You can also specify minimum and maximum capacity limits. Your desired capacity must be within the limit range.

**Desired capacity**

3 ↕

**Minimum capacity**

1

**Maximum capacity**

4

---

**Scaling policies - optional**

Choose whether to use a scaling policy to dynamically resize your Auto Scaling group to meet changes in demand. [info](#)

☐ Target tracking scaling policy  
 Choose a desired outcome and leave it to the scaling policy to add and remove capacity as needed to achieve that outcome.

☒ None

Selanjutnya next dan untuk tags di next aja lalu review kembali autocaling group jika sudah create

Selanjutnya create code Pipeline untuk CI/CD nya

Klik CodePipeline > Create Pipeline > isikan nama lalu next >

### Add source stage Info

**Source**

**Source provider**  
This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.

AWS CodeCommit

**Repository name**  
Choose a repository that you have already created where you have pushed your source code.

Sample-AngularApp

**Branch name**  
Choose a branch of the repository

master

**Change detection options**  
Choose a detection mode to automatically start your pipeline when a change occurs in the source code.

☒ **Amazon CloudWatch Events (recommended)**  
Use Amazon CloudWatch Events to automatically start my pipeline when a change occurs

☐ **AWS CodePipeline**  
Use AWS CodePipeline to check periodically for changes

Cancel Previous **Next**

Untuk Source Stage

Provider : AWS CodeCommit > pilih repository website angular > master branch > next

### Build - *optional*

**Build provider**  
This is the tool of your build project. Provide build artifact details like operating system, build spec file, and output file names.

AWS CodeBuild

**Region**

Europe (Ireland)

**Project name**  
Choose a build project that you have already created in the AWS CodeBuild console. Or create a build project in the AWS CodeBuild console and then return to this task.

or [Creating project](#)

**Environment variables - optional**  
Choose the key, value, and type for your CodeBuild environment variables. In the value field, you can reference variables generated by CodePipeline. [Learn more](#)

Add environment variable

Cancel Previous **Skip build stage** **Next**

Untuk Build

Provider : AWS CodeBuild > region default > project name belum ada maka di buat terlebih dahulu

## Create CodeBuild Project

**Project configuration**

Project name

A project name must be 2 to 255 characters. It can include the letters A-Z and a-z, the numbers 0-9, and the special characters - and \_.

Description - optional

► Additional configuration

tags

## Isikan Nama dan deskripsi

**Environment**

Environment image

☒ **Managed image**  
Use an image managed by AWS CodeBuild

☐ **Custom image**  
Specify a Docker image

Operating system

**i** The programming language runtimes are now included in the standard image of Ubuntu 18.04, which is recommended for new CodeBuild projects created in the console. See [Docker Images Provided by CodeBuild for details](#).

Runtime(s)

Image

Image version

Environment type

Privileged

☐ Enable this flag if you want to build Docker images or want your builds to get elevated privileges

Service role

☒ **New service role**  
Create a service role in your account

☐ **Existing service role**  
Choose an existing service role from your account

Role name

Type your service role name

Untuk Environment Image > Manage Image  
Operating System > Ubuntu



Image > 4.0 atau yang terbaru  
Image version > Last Version  
Environment Type > Ubuntu  
Service Role > New

▼ **Additional configuration**  
Timeout, certificate, VPC, compute type, environment variables, file systems

**Timeout**  
Default timeout is 1 hour

Hours:  Minutes:   
Timeout must be between 5 minutes and 8 hours

**Queued timeout**  
Default time in build queue is 8 hours

Hours:  Minutes:   
Timeout must be between 5 minutes and 8 hours

**Certificate**  
If you have a self-signed certificate or a certificate signed by a certification authority, choose the option to install it from your S3 bucket.

☒ Do not install any certificate ☐ Install certificate from your S3 bucket

**VPC**  
Select a VPC that your AWS CodeBuild project will access.

**Compute**

☒ 3 GB memory, 2 vCPUs  
☐ 7 GB memory, 4 vCPUs  
☐ 15 GB memory, 8 vCPUs  
☐ 145 GB memory, 72 vCPUs

**Environment variables**

Name	Value	Type	
<input type="text"/>	<input type="text"/>	Plaintext ▼	<input type="button" value="Remove"/>

**File systems**

Pada Bagian Additional Configuration

Bisa di lihat konfigurasi default nya dengan time 1 jam yang di mana waktu untuk mematika image ,untuk waktu antrianya 8 jam, setifikat jika menggunakan s3 Bucket bisa di pilih karena tidak menggunakan maka klik do not saja, pada computer terlihat pilihan memory dan cpu yang di gunakan,

Pada bagian ini biarkan saja default

### Buildspec

Build specifications

☒ Use a buildspec file  
Store build commands in a YAML-formatted buildspec file

☐ Insert build commands  
Store build commands as build project configuration

Buildspec name - *optional*

By default, CodeBuild looks for a file named buildspec.yml in the source code root directory. If your buildspec file uses a different name or location, enter its path from the source root here (for example, buildspec-two.yml or configuration/buildspec.yml).

### Logs

CloudWatch

☒ CloudWatch logs - *optional*  
Checking this option will upload build output logs to CloudWatch.

Group name

Stream name

S3

☐ S3 logs - *optional*  
Checking this option will upload build output logs to S3.

Untuk BuildSpec > use buildspec file : karena kita telah menyiapkan file spesifikasi yang akan di jalan oleh image lalu klik continue

Kita kembali lagi ke halaman build pipeline dan terlihat code build yang di buat tadi pada pilihan projectname nya lalu next

### Build - *optional*

Build provider

This is the tool of your build project. Provide build artifact details like operating system, build spec file, and output file names.

AWS CodeBuild

Region

Europe (Ireland)

Project name

Choose a build project that you have already created in the AWS CodeBuild console. Or create a build project in the AWS CodeBuild console and then return to this task.

myangularbuild02

×

 or 

Create project

Successfully created myangularbuild02 in CodeBuild.

×

Environment variables - *optional*

Choose the key, value, and type for your CodeBuild environment variables. In the value field, you can reference variables generated by CodePipeline. [Learn more](#)

Add environment variable

Cancel

Previous

Skip build stage

Next

Selanjutnya Creat Applicaton di CodeDeploy

Pilih CodeDeploy > Create Application

### Create application

**Application configuration**

Application name  
Enter an application name

100 character limit

Compute platform  
Choose a compute platform

Cancel Create application

Isi nama dan pilih compute platform EC2On-Preasmises > Create Application

**Application**

Application  
AngularApp2  
Compute type  
EC2/On-premises

**Deployment group name**

Enter a deployment group name

100 character limit

**Service role**

Enter a service role  
Enter a service role with CodeDeploy permissions that grants AWS CodeDeploy access to your target instances.

Isikan nama group deployment

Service role pilih service role yang telah di buat sebelumnya

## Deployment type

Choose how to deploy your application

☐ In-place

Updates the instances in the deployment group with the latest application revisions. During a deployment, each instance will be briefly taken offline for its update

☒ Blue/green

Replaces the instances in the deployment group with new instances and deploys the latest application revision to them. After instances in the replacement environment are registered with a load balancer, instances from the original environment are deregistered and can be terminated.

Untuk Deployment type > Blue/Green

## Environment configuration

Specify the Amazon EC2 Auto Scaling groups or Amazon EC2 instances where the current application revision is deployed.

☒ Automatically copy Amazon EC2 Auto Scaling group

Provision an Amazon EC2 Auto Scaling group and deploy the new application revision to it. AWS CodeDeploy will create the Auto Scaling group by copying the one you specify here.

☐ Manually provision instances

I will specify here the instances where the current application revision is running. I will specify the instances for the replacement environment when I create a deployment.

Choose the Amazon EC2 Auto Scaling group where the current application revision is deployed.

🔍 My-Auto-Scaling-Group-2



Pada Environment > Autoscalling > pilih autscaling group yang di buat sebelumnya

### Deployment settings

**Traffic rerouting**

☒ Reroute traffic immediately  
☐ I will choose whether to reroute traffic

---

Choose whether instances in the original environment are terminated after the deployment is succeeds, and how long to wait before termination.

☒ Terminate the original instances in the deployment group  
☐ Keep the original instances in the deployment group running

Days:  ▼     
 Hours:  ▼     
 Minutes:  ▼

---

**Deployment configuration**

Choose from a list of default and custom deployment configurations. A deployment configuration is a set of rules that determines how fast an application is deployed and the success or failure conditions for a deployment.

▼    or   

Pada Deployment setting

Traffic routing biarkan Route Traffic Immediately

Untuk waktu terminate original instance yang sebelumnya 1 jam ganti menjadi 15 menit

Deploymnet Congfiguration pilih CodeDeployDefaultAllatOne

### Load balancer

Select a load balancer to manage incoming traffic during the deployment process. The load balancer blocks traffic from each instance while it's being deployed to and allows traffic to it again after the deployment succeeds.

☒ Enable load balancing

☐ Application Load Balancer or Network Load Balancer     
 ☒ Classic Load Balancer

Choose a load balancer

▼

Pada Load Balancer

Pilih Classic Loacbalancer yang telah di buat sebelumnya

### Rollbacks

Enable deployment rollbacks for this deployment group

☒ Roll back when a deployment fails

☐ Roll back when alarm thresholds are met

☐ Disable rollbacks

Untuk Menambahkan Fitur Rollback pilih rollback when a deployment fails  
Selanjutnya klik CreateDeploymentGroup

#### Deployment group details

Deployment group name AngularApp-Auto-Scaling	Application name <a href="#">AngularApp2</a>	Compute platform EC2/On-premises
Deployment type Blue/green	Service role ARN arn:aws:iam::837881029860:role/CodeDeployEC2ServiceRole2	Deployment configuration <a href="#">CodeDeployDefault.AllAtOnce</a>
Rollback enabled True	Agent update scheduler <a href="#">Learn to schedule update in AWS Systems Manager</a>	

#### Environment configuration: Amazon EC2 Auto Scaling groups

Name
My-Auto-Scaling-Group-2

Ini adalah summary dari DeploymentGroup yang di buat

Kita kembali lagi Ke pembuatan CodePipeline

### Deploy - optional

Deploy provider  
Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.

AWS CodeDeploy

Region  
Europe (Ireland)

Application name  
Choose an application that you have already created in the AWS CodeDeploy console. Or create an application in the AWS CodeDeploy console and then return to this task.

AngularApp2

Deployment group  
Choose a deployment group that you have already created in the AWS CodeDeploy console. Or create a deployment group in the AWS CodeDeploy console and then return to this task.

AngularApp-Auto-Scaling

Cancel Previous Skip deploy stage Next

Untuk deployment provider > AWS CodeDeploy  
Region > Default  
Application name : pilih aplikasi yang di buat sebelumnya



Deployment Group : Pilih yang sebelumnya di buat  
Next

### Step 1: Choose pipeline settings

#### Pipeline settings

Pipeline name  
Angular-Blue-Green-Pipeline2  
Artifact location  
codepipeline-eu-west-1-624450656211  
Service role name  
AWSCodePipelineServiceRole-eu-west-1-Angular-Blue-Green-Pipelin

### Step 2: Add source stage

#### Source action provider

Source action provider  
AWS CodeCommit  
RepositoryName  
Sample-AngularApp  
BranchName  
master  
PollForSourceChanges  
false

### Step 3: Add build stage

#### Build action provider

Build action provider  
AWS CodeBuild  
ProjectName  
myangularbuild02

### Step 4: Add deploy stage

#### Deploy action provider

Deploy action provider  
AWS CodeDeploy  
ApplicationName  
AngularApp2  
DeploymentGroupName  
AngularApp-Auto-Scaling

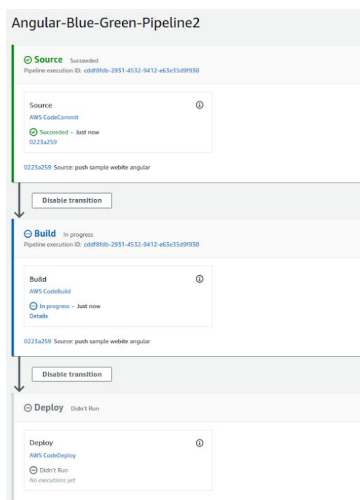
Cancel

Previous

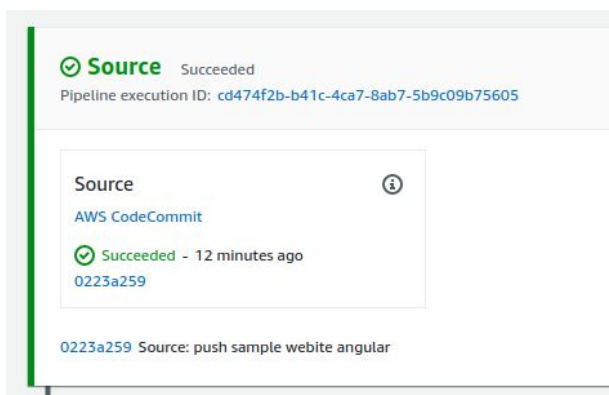
Create pipeline

Ini adalah review dari piple yang di buat lalu klik create pipeline

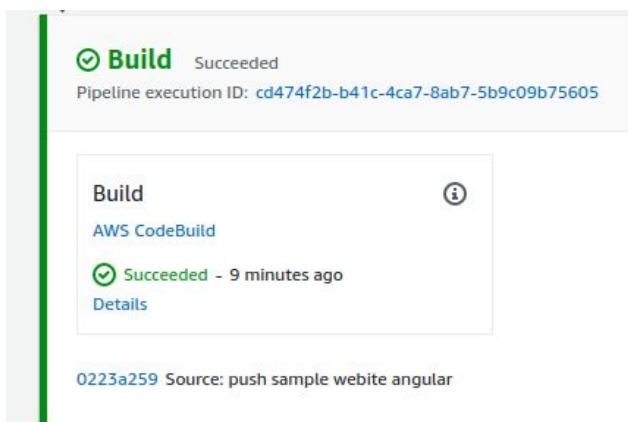
## Proses Deployment Berjalan



## Pertama



Source Code akan di Ambil dari Repository CodeCommit  
Dan terlihat commit id dan deksipis nya



Kedua Proses Build yang di mana ini akan membuild image yang telah di bikin sebelumnya sesuai dengan script yang ada di repository "buildspec.yml"

```

! buildspec.yml ×
! buildspec.yml
1  version: 0.2
2  phases:
3    install:
4      runtime-versions:
5        nodejs: 12
6      commands:
7        - npm install -g @angular/cli@9.0.6
8    pre_build:
9      commands:
10       - npm install
11    build:
12      commands:
13        - ng build --prod
14      finally:
15        - echo This is the finally block execution!
16  artifacts:
17    files:
18      - 'dist/my-angular-project/**/*'
19      - appspec.yml
20      - 'deploy-scripts/**/*'

```

Dengan versi 0.2 dan runtime nodejs12 lalu install angular cli  
 Pada pre\_build menginstall npm sebagai dependensi untuk menjalankan angular  
 Build yaitu proses membuild atau menjalankan angular  
 Untuk artifak definisikan tempat atau file dimana angular file berada

```

! appspec.yml
1  version: 0.0
2  os: linux
3  files:
4    - source: dist/my-angular-project
5      destination: /var/www/my-angular-project
6  permissions:
7    - object: /var/www/my-angular-project
8      pattern: '*'
9      mode: '0755'
10     owner: root
11     group: root
12     type:
13       - file
14       - directory
15  hooks:
16    ApplicationStart:
17      - location: deploy-scripts/application-start-hook.sh
18        timeout: 300
19    #ValidateService:
20      # - location: deploy-scripts/simulate-failure.sh

```

dan berdasarkan spesifikasi webserver yang di buat pada AIMS image/Code Deploy Agent sebelumnya, yang menggunakan os Linux sorce pada dirctory my-angular-project yang berada pada directory /var/www/my-angular-project, untuk permission directory “755” yang semuanya root, untuk hooks ini menjalankan file application-start-hook.sh yang isinya untuk menrestart webserver dengan mode shell script

```
application-start-hook.sh X
deploy-scripts > application-start-hook.sh
1  #!/bin/bash
2
3  sudo service nginx restart
```

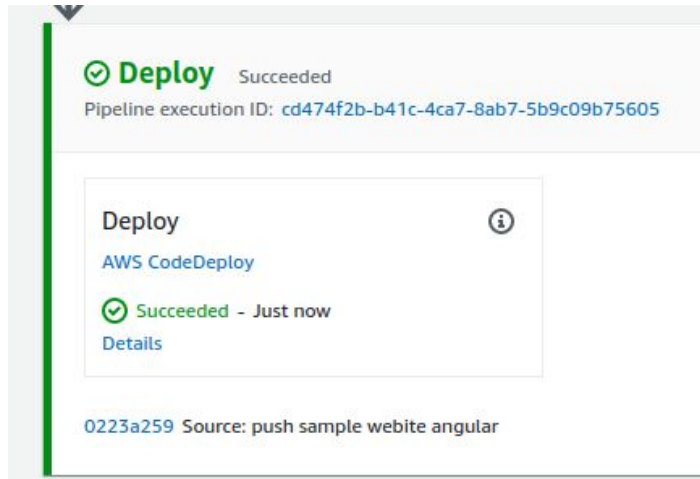
Untuk lebih detail apa saja yang di kerjakan

```
1 [Container] 2020/07/13 07:14:54 Waiting for agent ping
2 [Container] 2020/07/13 07:14:56 Waiting for DOWNLOAD_SOURCE
3 [Container] 2020/07/13 07:14:57 Phase is DOWNLOAD_SOURCE
4 [Container] 2020/07/13 07:14:57 CODEBUILD_SRC_DIR=/codebuild/output/src868757086/src
5 [Container] 2020/07/13 07:14:57 YAML location is /codebuild/output/src868757086/src/buildspec.yml
6 [Container] 2020/07/13 07:14:57 Processing environment variables
7 [Container] 2020/07/13 07:14:57 Selecting 'nodejs' runtime version '12' based on manual selections...
8 [Container] 2020/07/13 07:14:57 Running command echo "Installing Node.js version 12 ..."
9 Installing Node.js version 12 ...
10
11 [Container] 2020/07/13 07:14:57 Running command n $NODE_12_VERSION
12 installed : v12.18.0 (with npm 6.14.4)
13
14 [Container] 2020/07/13 07:15:07 Moving to directory /codebuild/output/src868757086/src
15 [Container] 2020/07/13 07:15:07 Registering with agent
16 [Container] 2020/07/13 07:15:07 Phases found in YAML: 3
17 [Container] 2020/07/13 07:15:07 INSTALL: 1 commands
18 [Container] 2020/07/13 07:15:07 PRE_BUILD: 1 commands
19 [Container] 2020/07/13 07:15:07 BUILD: 1 commands
20 [Container] 2020/07/13 07:15:07 Phase complete: DOWNLOAD_SOURCE State: SUCCEEDED
21 [Container] 2020/07/13 07:15:07 Phase context status code: Message:
22 [Container] 2020/07/13 07:15:07 Entering phase INSTALL
23 [Container] 2020/07/13 07:15:07 Running command npm install -g @angular/cli@9.0.6
24 npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142
25 /usr/local/bin/ng -> /usr/local/lib/node_modules/@angular/cli/bin/ng
26
27 > @angular/cli@9.0.6 postinstall /usr/local/lib/node_modules/@angular/cli
28 > node ./bin/postinstall/script.js
```

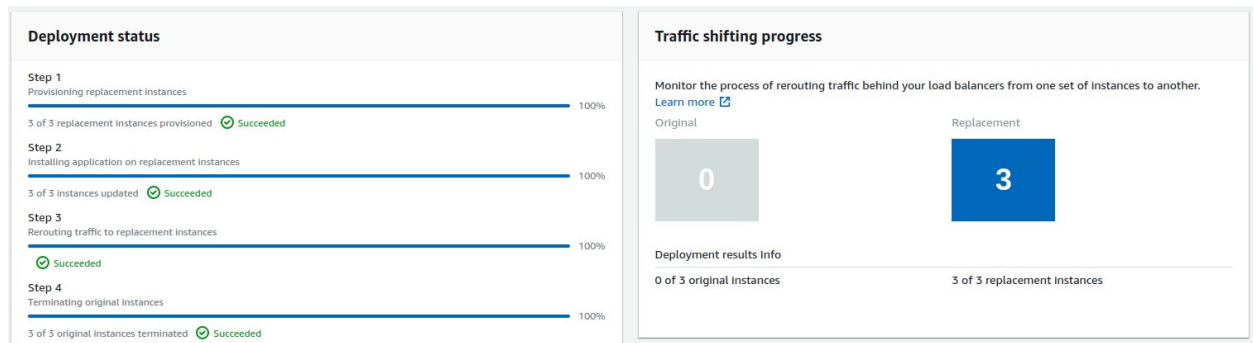
Di sini terlihat codebuild menjalan contaner sesuai dengan script dari buildspec.yml  
Untuk melihat fase2 pemBuild bisa klik deatil > phase detail

Name	Status	Context	Duration	Start time	End time
SUBMITTED	⊙ Succeeded	-	<1 sec	Jul 13, 2020 2:14 PM (UTC+7:00)	Jul 13, 2020 2:14 PM (UTC+7:00)
QUEUED	⊙ Succeeded	-	1 sec	Jul 13, 2020 2:14 PM (UTC+7:00)	Jul 13, 2020 2:14 PM (UTC+7:00)
PROVISIONING	⊙ Succeeded	-	19 secs	Jul 13, 2020 2:14 PM (UTC+7:00)	Jul 13, 2020 2:14 PM (UTC+7:00)
DOWNLOAD_SOURCE	⊙ Succeeded	-	10 secs	Jul 13, 2020 2:14 PM (UTC+7:00)	Jul 13, 2020 2:15 PM (UTC+7:00)
INSTALL	⊙ Succeeded	-	13 secs	Jul 13, 2020 2:15 PM (UTC+7:00)	Jul 13, 2020 2:15 PM (UTC+7:00)
PRE_BUILD	⊙ Succeeded	-	37 secs	Jul 13, 2020 2:15 PM (UTC+7:00)	Jul 13, 2020 2:15 PM (UTC+7:00)
BUILD	⊙ Succeeded	-	95 secs	Jul 13, 2020 2:15 PM (UTC+7:00)	Jul 13, 2020 2:17 PM (UTC+7:00)
POST_BUILD	⊙ Succeeded	-	<1 sec	Jul 13, 2020 2:17 PM (UTC+7:00)	Jul 13, 2020 2:17 PM (UTC+7:00)
UPLOAD_ARTIFACTS	⊙ Succeeded	-	<1 sec	Jul 13, 2020 2:17 PM (UTC+7:00)	Jul 13, 2020 2:17 PM (UTC+7:00)
FINALIZING	⊙ Succeeded	-	2 secs	Jul 13, 2020 2:17 PM (UTC+7:00)	Jul 13, 2020 2:17 PM (UTC+7:00)
COMPLETED	⊙ Succeeded	-	-	Jul 13, 2020 2:17 PM (UTC+7:00)	-

Selanjutnya ke proses ke tiga yakni Build



Untuk melihat lebih detail klik detail



Terlihat status dari deployment nya ada 4 step

Step Pertama : mereplace instance yang telah ada berjumlah 3 sesuai dengan yang di buat sebelumnya

Step kedua : menginstall aplikasi ke instance yang di replacement

Step ke tiga : mengarahkan trafik dari instance original ke instance replace

Step ke empat : mematikan instance yang original

Dan di sampingnya juga terlihat trafik perpindahan dari instance original ke instance replacement

i-0042d98a431aa42...	t2.micro	eu-west-1c	running	2/2 checks ...	None	ec2-3-249-154-23.eu-w...	3.249.154.23	-	aws-ec2	disabled
i-00802a65412b617...	t2.micro	eu-west-1c	running	2/2 checks ...	None	ec2-34-244-244-108.eu...	34.244.244.108	-	aws-ec2	disabled
i-00b2a1b90de241d...	t2.micro	eu-west-1c	terminated		None		-	-	aws-ec2	disabled
i-02b32ccac3f456c7a	t2.micro	eu-west-1b	running	2/2 checks ...	None	ec2-34-245-190-33.eu-...	34.245.190.33	-	aws-ec2	disabled
i-04949bb0c278157a	t2.micro	eu-west-1a	terminated		None		-	-	aws-ec2	disabled
i-0667387b5f999866c	t2.micro	eu-west-1b	terminated		None		-	-	aws-ec2	disabled

Untuk memastikan instance yang di replacement bisa di dilat pada dashboard EC2

Di sini terlihat ada 3 instance yang terminate/mati dan 3 inatance yang running



Untuk memastikan apakah angular web berjalan atau tidak, masuk loadbalacer yang di buat sebeumnya

Load balancer: ELB-C

Description Instances Health check Listeners Monitoring Tags Migration

Basic Configuration

Name	ELB-C	Creation time	July 13, 2020 at 1:55:31 AM UTC+7
* DNS name	ELB-C-1624424059.eu-west-1.elb.amazonaws.com (A Record)	Hosted zone	Z32012XQLNTSW2
Type	Classic (Migrate Now)	Status	3 of 3 instances in service
Scheme	internet-facing	VPC	vpc-7c0ff605
Availability Zones	subnet-91595cd9 - eu-west-1a, subnet-9f3245c5 - eu-west-1b, subnet-f40c2692 - eu-west-1c		

Port Configuration

Copy DNS name ke URL browser untuk mastikanya

Not secure | elb-c-1624424059.eu-west-1.elb.amazonaws.com

## Angular App Untuk Implementasi Code Deploy Untuk Blue Green Deployment CI/CD Code Commit

Version: 1.0

Selamat Telah Berhasil Deploy Menggunakan CodeDeploy AWS.

Angular 9 and Bootstrap 4.4.1 contoh AWS CodeDeploy

### Simple Calculator

Your first input

Please select an operator

Your second input

Clear

Calculate

Dan terlihat web aplikasinya sudah terdeploy / berjalan

Untuk memastikan ada berapa instance yang di arakah melalui DNS name yang ada di LoadBalancer klik instctance dan terlihat ada 3 instance yang di arahakan oleh DNS name tersebut.

Description Instances Health check Listeners Monitoring Tags Migration

Connection Draining: Enabled, 300 seconds (Edit)

Edit Instances

Instance ID	Name	Availability Zone	Status	Actions
i-0042d98a431aa4221		eu-west-1c	InService ⓘ	<a href="#">Remove from Load Balancer</a>
i-0a72585f89a3ec668		eu-west-1a	InService ⓘ	<a href="#">Remove from Load Balancer</a>
i-02b32ccac3f456c7a		eu-west-1b	InService ⓘ	<a href="#">Remove from Load Balancer</a>